1. Non-Healthcare Industries Panel

- When you first implemented standards-based interoperability across organizational boundaries in your industry, what business problem were you trying to solve? What standards did you use and why? What were the outcomes you were looking for? Were those outcomes achieved?

With ODBC, was trying to make it possible for any client tool to access any database. Built a killer application, Access to showcase this, build the test platform to test for conformance, and then got other vendors to support it. The outcomes were achieved.

With AJAX/HTML was tying to make it possible to build richer applications on the Web which responded in a more dynamic way to user actions. Build IE 4. And then IE 5.0 and XML Microsoft components to showcase it along with demos. Outcome achieved.

With XML was trying to make it possible for any application on the web to exchange data with any other application. Used IE 4/5 to showcase this technology. Also sold the idea to IBM and Oracle and others to drive industry adoption. Outcome achieved.

With Google API's was trying to make it really easy to read google application's data such as appointments in a calendar leveraging existing standards. Used RSS as the basic building block for this because many applications already showcased this technology and its use was widely understood. Outcome mixed.

- How did you phase in your interoperability standards over time? What did you do to encourage adoption?

To encourage adoption, build testing platforms for all, kept them simple and focused where possible, got books written on how to use them, had working applications with them at all times, and taught users them at conferences for normal programmers. All standards took about 6 months to 12 months to be properly interoperable except for AJAX which is still not as interoperable between IE and Firefox and Safari as one would like.

- Did implementing interoperability in your industry help you achieve your goals, or did it inhibit progress toward achieving your goals? What role did the standards play, what was the rate of adoption and the impact on overall costs?

Interoperability was key for my goals for most of these standards. They were designed to exchange data or provide a common programming model across millions of engineers.

- Can you reflect on which types of standards were most successful in achieving widespread adoption across your industry and which ones were less successful and why?

I have a post I'm going to publish about this but essentially successful standards are ones that are simple, small, focused, and easy for programmers to understand and build, and have real applications during the process of defining the standard or are derived from experience in building real applications.

- Was there significant asymmetry in your sector in terms of IT and infrastructure capability? If so, what impact did this have?

Not in my sector.

- What is an example of your greatest success and your most frustrating issue from the implementation?

XML was my greatest success. My most frustrating issue was XML Schema which became the antithesis of everything a good standard is.

- What advice would you give to healthcare to help them as an industry mitigate problems or accelerate adoption of interoperable health information technology in order to improve quality and cost effectiveness?

Start simple. Focus on exchanging key data (conditions, test results, medicines, and demographics and family history for conditions). Make it computable at all times. That means codes, not text. Don't try to boil the ocean and solve all interoperability issues. Encourage/reward use of standard encodings for the doctor, practice, patient id, and conditions where standards are weakest.

- Can you give an example of an instance where interoperability led to rapid transformation of a business process in your sector? If so, what were the characteristics of the standards used and the adoption rates that led to this transformation?

XML is such an example. There was a huge need for applications to work with others across the web as the web moved us from computing islands to connected worlds and do so swiftly, easily, and without custom development done in tandem between the applications. This is actually very similar to our health care environment today.